



# Tech Info Library

## **MacsBug 6.5.2: ReadMe File (8/95)**

Article Created: 09 August 1995

TOPIC -----

This article contains the Read Me file included with MacsBug 6.5.2

DISCUSSION -----

About MacsBug 6.5.2  
7/26/95

With the 6.5.1 release, we've adopted a newer, hopefully less-confusing version strategy. From now on, each new version of MacsBug will be a "final" version. Thus, the next version will be 6.5.3, the next bug-fix version to 6.5.

•• NOTE •• - MacsBug 6.5.1 was never released, so please be sure to read what changed in that version since it's not repeated here, but is significant.

### MIXED MODE STEPPING

You can now step across all Mixed Mode transitions. This means that when you come up to a MixedModeMagic trap, stepping once more lands on the first instruction on the other side, whether it's 68K or PowerPC. Similarly, when you're stepping through PowerPC code that calls the functions CallUniversalProc or CallOSTrapUniversalProc, you will automatically land on the other side of the call. When returning from a 68K routine to PowerPC code or vice-versa, stepping the return instruction will take you back to the caller, whether it's 68K or PowerPC. There should be no instances where stepping takes you into Mixed Mode itself (so if you're debugging Mixed Mode, use another debugger).

Mixed Mode stepping works equally well with accelerated ("fat") resources. If your 68K code loads and calls into a fat resource, you can simply step right into it.

### NMIs IN NATIVE CODE

Using the NMI key sequence, you can now interrupt native code execution, and will land on the PowerPC instruction you interrupted. However, due to reentrancy concerns, you cannot interrupt in the middle of Mixed Mode execution, you'll land in 68K code like you used to do with older versions of MacsBug.

## BUS ERROR CATCHING ON POWERPC

Bus error handling on the first-generation PowerPC machines (6100/7100/8100) now works. This really wasn't a bug in MacsBug, but I fixed it anyway. The problem was that the sound-in component trashes the bus error vector when it is loaded (this is the case in 7.5 and 7.5.1). MacsBug now takes care of the component load through a magical patch, and fixes up the bus error vector if it becomes damaged. See, MacsBug really is system software.

## POWERPC STACK CRAWL

Fixed the infinite loop bug in the "SC" command when walking back over PowerPC->68K switch frames (mentioned in the 6.5.1 release notes).

## POWERPC DEBUGGER/DEBUGSTR CALLS

When you execute a Debugger or DebugStr in native code, you should no longer land in the implementation (in InterfaceLib at present), you'll be in your own code right after the branch to the library where they live.

## POWERPC STEPPING

Stepping or tracing with a count now works in PowerPC code.

## POWERPC REGISTER DISPLAY

When on a PowerPC machine, we now show the current interrupt mask in place of the Memory Manager mode in the left sidebar area (you're always in 32-bit mode). This is shown (and is correct) when in both 68K or PowerPC code.

The current interrupt mask is shown in the PowerPC register set when the "TD" command is used.

Changed "TF" command to show the exponent and sign of each FPR as one component.

## REGISTER ASSIGNMENT

You can now set and get A5 when in both 68K and PowerPC code. For the moment, this is the only 68K register you can get to since it's the most useful. This means that macros like "thecport" work regardless of the current instruction set.

The 68020-040 register formerly known as "DFC" (Data Function Code) must now be called "RDFC" so it's no longer hex (nobody cares about this register anyway).

## COMMANDS FIXED/ADDED

Added a new command "STAT" and incorporated it into the StdLog macro. It shows lots of useful information you can see on the screen, but have never had an easy way of getting into the log (at least not very concisely). For instance, it identifies the names of all shipping machines that have unique box flag values,

shows used stack space, max-used stack space when available, and ResErr/MemErr when they are nonzero.

The "EA" (Exit to Application) and "ES" (Exit to Shell) commands now work when you're in PowerPC code.

The "IR" (Instruction list to Return) command now works in PowerPC code with no address.

"IRP" (Instruction list to Return PowerPC) command has been added.

"GTP" (Go Till in PowerPC code) command has been added.

Restarting (RS/RB) now works in a general way on PCI-based machines. It used to need a box flag table, which was a work-around for early developmental ROMs without a reasonable Gestalt selector to identify that the machine needed a hard reset to restart.

The "DMA" (Display Memory ASCII) command now honors the DITTO mode.

#### WH COMMAND

When displaying the CFM library an address lives in, we now display the executable and writable status of the section the address is in (if any).

If an address is in low memory, we now say so (instead of saying it's in RAM but not in a know heap).

#### A-TRAP RECORDING ON QUADRA/PERFORMA 63x AND POWERBOOK 150

It works, and doesn't make your machine crash. I really don't want to tell you why 'cause it would make a great puzzle page.

#### A-TRAP ACTIONS

A-Trap tracing and recording no longer mask trap values into the A000-A0FF and A800-ABFF range when displaying them, all bits are shown as-is when the trap was executed.

A-Trap trace now shows the name of the real call that was made if the trap is selector-based (dispatched), rather than the trap. This means you'll see traces with information such as "SameProcess" instead of "OSDispatch", which is much more useful.

A-Trap recording and tracing now identifies Enqueue and Dequeue as stack-based traps rather than register-based (they aren't).

A-Trap break, trace, and step-spy logic has been fixed so you can now have combinations of them pending at the same time that actually work. In the past you couldn't have a trace and a break work at the same time, the first one that was activated got precedence. This means you can now have multiple trace ranges and they'll work as well.

Fixed the long-standing (since November 24, 1992, to be exact) bug where you'd land near a TRAP # $\$D$  inside MacsBug if you stepped over an A-Trap that had a break action set on it.

Setting an A-Trap break while in PowerPC code works (I was stupid and failed to install the A-Trap dispatcher patch).

#### 68K DISASSEMBLY

When disassembling a MixedModeMagic trap (AAFE) that's at the current 68K PC, we now show the address that you'll land if you step across the mode transition, as well as the destination instruction set.

When disassembling an A-Trap instruction, the comment field now contains the implementation address of the trap, rather than the trap word. It's already shown off to the right anyway.

#### POWERPC DISASSEMBLY

When disassembling PowerPC code that uses an offset off of R0, we now look up macros for that address. For example, instead of getting this disassembly:

```
lwz r3, 0x064F(0)
```

you'll get:

```
lwz r3, JFlushCache(r0)
```

Throughout PowerPC disassembly all fields are now shown in hex instead of decimal.

The symbol lookup code now lets tracebacks win if they're at the same address as a previously found CFM export. This lets the disassembler munch the embedded data making the disassembly look nicer (and smaller).

The glue sniffer can now identify Microsoft-style cross-TOC glue.

#### SYMBOL LOOKUP

I drastically sped up CFM/traceback lookups over 6.5.1 so MacsBug is usable again.

Optimized embedded 68K symbol lookup so it's about 10% faster.

#### MISCELLANEOUS

Fixed all instances (hopefully) where switching ISAs while in MacsBug wouldn't cause the register names to be erased and redrawn. My internal concept of current and previous ISA was a bit loopy.

The syntax error display for all non-repeating commands has been fixed (br, etc.) You'll no longer see the entire help text as one garbled mess. Yeah, it

confused me too.

The "explain significance of" feature shows nicer output for slightly-negative numbers (0 to -16M).

The display of "is a bad pointer" only yells about odd pointers on 68000s.

Made internal commands override dcmds of the same name. You can specify a dcmd (only) by putting a "[option-d]" in front of the name (and it should work for Help, too, as in "[option-d]cfm").

Allow "[option-=]" for not-equal (Dave did it, I'll never remember it).

Moved the typographical quotes up by one pixel (to make Greg Robbins happy).

#### BUILT-IN DEBUGGER PREFS

Fixed GetlXType and GetlXResource to be GetlIndType and GetlIndResource (we've added two macros to let you still type GetlXType and GetlXResource if you really want to).

Replaced PPCJump macro with a more general one. PPCJumpA is now the old "branch absolute" version.

Copyright 1995 Apple Computer, Inc.

Support Information Services  
Copyright 1995, Apple Computer, Inc.

Keywords: <None>

=====

This information is from the Apple Technical Information Library.

19960215 11:05:19.00

Tech Info Library Article Number: 18348